<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example.employee</groupId>

<artifactId>employee-webapp</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Servlet API -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!-- MySQL JDBC Driver -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>3.3.1</version>

<configuration>

<warSourceDirectory>WebContent</warSourceDirectory>

</configuration>

</plugin>

</plugins>

</build>

</project>

package com.example.employee.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBUtil {

public static Connection getConnection() throws SQLException {

String url = "jdbc:mysql://localhost:3306/your\_database";

String user = "your\_username";

String password = "your\_password";

return DriverManager.getConnection(url, user, password);

}

}

package com.example.employee.model;

public class Employee {

private int id;

private String name;

private String email;

private String department;

public Employee(int id, String name, String email, String department) {

this.id = id;

this.name = name;

this.email = email;

this.department = department;

}

// Getters and setters

public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getEmail() { return email; }

public void setEmail(String email) { this.email = email; }

public String getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

}

package com.example.employee.dao;

import com.example.employee.model.Employee;

import com.example.employee.util.DBUtil;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class EmployeeDAO {

public void addEmployee(Employee employee) {

String query = "INSERT INTO employee (name, email, department) VALUES (?, ?, ?)";

try (Connection conn = DBUtil.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, employee.getName());

stmt.setString(2, employee.getEmail());

stmt.setString(3, employee.getDepartment());

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

public List<Employee> getAllEmployees() {

List<Employee> employees = new ArrayList<>();

String query = "SELECT \* FROM employee";

try (Connection conn = DBUtil.getConnection();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

while (rs.next()) {

Employee employee = new Employee(

rs.getInt("id"),

rs.getString("name"),

rs.getString("email"),

rs.getString("department")

);

employees.add(employee);

}

} catch (SQLException e) {

e.printStackTrace();

}

return employees;

}

public void updateEmployee(Employee employee) {

String query = "UPDATE employee SET name=?, email=?, department=? WHERE id=?";

try (Connection conn = DBUtil.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setString(1, employee.getName());

stmt.setString(2, employee.getEmail());

stmt.setString(3, employee.getDepartment());

stmt.setInt(4, employee.getId());

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

public void deleteEmployee(int id) {

String query = "DELETE FROM employee WHERE id=?";

try (Connection conn = DBUtil.getConnection();

PreparedStatement stmt = conn.prepareStatement(query)) {

stmt.setInt(1, id);

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

package com.example.employee.controller;

import com.example.employee.dao.EmployeeDAO;

import com.example.employee.model.Employee;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.\*;

import java.io.IOException;

import java.util.List;

@WebServlet("/employee")

public class EmployeeController extends HttpServlet {

private EmployeeDAO employeeDAO = new EmployeeDAO();

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String action = request.getParameter("action");

if ("add".equals(action)) {

request.getRequestDispatcher("addEmployee.jsp").forward(request, response);

} else if ("list".equals(action)) {

List<Employee> employees = employeeDAO.getAllEmployees();

request.setAttribute("employees", employees);

request.getRequestDispatcher("employeeList.jsp").forward(request, response);

}

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String name = request.getParameter("name");

String email = request.getParameter("email");

String department = request.getParameter("department");

Employee employee = new Employee(0, name, email, department);

employeeDAO.addEmployee(employee);

response.sendRedirect("employee?action=list");

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Employee Management System</title>

</head>

<body>

<h1>Employee Management</h1>

<a href="employee?action=list">View Employees</a> |

<a href="employee?action=add">Add Employee</a>

</body>

</html>

<web-app xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_4\_0.xsd"

version="4.0">

<servlet>

<servlet-name>EmployeeController</servlet-name>

<servlet-class>com.example.employee.controller.EmployeeController</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>EmployeeController</servlet-name>

<url-pattern>/employee</url-pattern>

</servlet-mapping>

</web-app>